(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2002 (07.03.2002)

PCT

(10) International Publication Number
WO 02/019064 A3

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/CA01/01239

(22) International Filing Date: 31 August 2001 (31.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/229,859 1 September 2000 (01.09.2000) US

(71) Applicant and
(72) Inventor: **BUCKLEY, Conleth** [CA/CA]; 2651 Colman Street, Ottawa, Ontario K1V 8J7 (CA).

(74) Agent: **SADIK, Achmed, N.**; P.O. Box 908 Station B, Ottawa, Ontario K1P 5P9 (CA).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

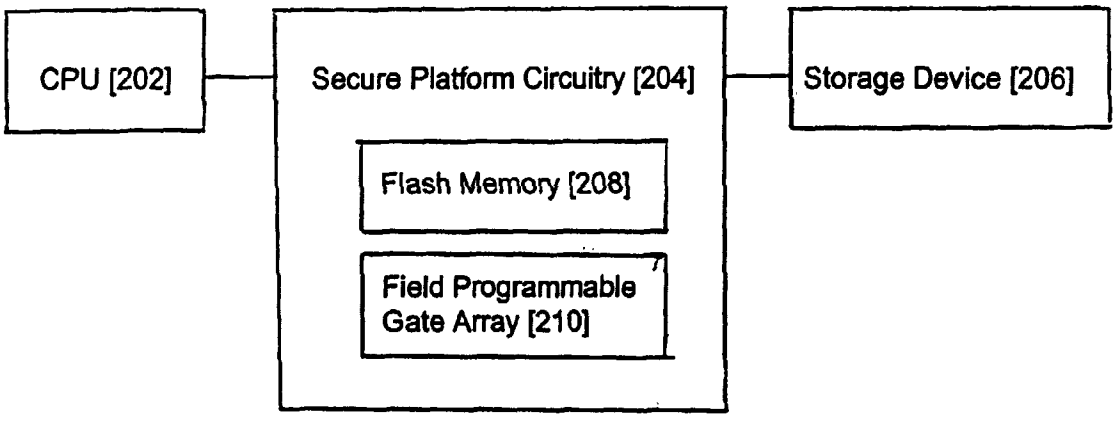
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
24 April 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR PREVENTING UNWANTED ALTERATIONS OF DATA AND PROGRAMS STORED IN A COMPUTER SYSTEM



(57) Abstract: The present invention relates to a method and system for preventing the unwanted alteration of data and programs stored within a computer system. The system employs a field programmable gate array to control access to a storage device. Different profiles can be accessed through the use of passwords. Different profiles provide different control parameters for access to the storage device. The gate array can be reprogrammed from time to time using downloadable electronic files. Security is achieved in the download by using keys and encryption techniques.

WO 02/019064 A3

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 586 301 A (FISHERMAN IGOR ET AL) 17 December 1996 (1996-12-17) column 1, line 15 - line 37 column 2, line 29 - line 43 column 3, line 30 -column 4, line 10 column 5, line 50 -column 6, line 9 column 6, line 29 - line 36 column 7, line 43 -column 8, line 2 column 8, line 46 - line 59 figures 1,2,4 ---	1,2
X	EP 0 949 556 A (FUJITSU LTD) 13 October 1999 (1999-10-13) paragraph '0003! paragraphs '0025!,'0026! paragraph '0050! paragraphs '0052!,'0053! figures 1,2,4,5 --- -/--	1

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 January 2003	31/01/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Arbutina, L

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 945 775 A (FUJITSU LTD) 29 September 1999 (1999-09-29) paragraphs '0016!,'0017! paragraph '0023! paragraph '0030! figures 1,2,5,12	1
X	WO 99 21094 A (QUICKFLEX INC ;LEDZIUS ROBERT C (US)) 29 April 1999 (1999-04-29) page 3, line 3 -page 5, line 6 page 24, line 19 -page 28, line 3 figure 8	3,4
X	WO 99 56428 A (MOTOROLA INC) 4 November 1999 (1999-11-04) the whole document	3,4
A	EP 0 851 358 A (SUN MICROSYSTEMS INC) 1 July 1998 (1998-07-01) abstract column 1, line 1 - line 24 column 2, line 17 - line 27 column 2, line 48 - line 54 column 3, line 29 - line 42 column 4, line 14 - line 27 figures 1,3	1-4

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5586301 A	17-12-1996	AU 4129796 A	06-06-1996
		CA 2204860 A1	23-05-1996
		EP 0792484 A1	03-09-1997
		WO 9615486 A1	23-05-1996
		US 5657470 A	12-08-1997
EP 0949556 A	13-10-1999	JP 11296436 A	29-10-1999
		EP 0949556 A2	13-10-1999
EP 0945775 A	29-09-1999	JP 11265544 A	28-09-1999
		EP 0945775 A2	29-09-1999
WO 9921094 A	29-04-1999	CA 2308755 A1	29-04-1999
		EP 1025503 A2	09-08-2000
		TW 456103 B	21-09-2001
		WO 9921094 A2	29-04-1999
WO 9956428 A	04-11-1999	US 6141756 A	31-10-2000
		AU 3183499 A	16-11-1999
		BR 9906398 A	26-09-2000
		CN 1266571 T	13-09-2000
		EP 0990326 A1	05-04-2000
		JP 2002507307 T	05-03-2002
		WO 9956428 A1	04-11-1999
EP 0851358 A	01-07-1998	US 5911778 A	15-06-1999
		EP 0851358 A2	01-07-1998
		JP 10228420 A	25-08-1998

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2002 (07.03.2002)

PCT

(10) International Publication Number
WO 02/19064 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/CA01/01239

(22) International Filing Date: 31 August 2001 (31.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/229,859 1 September 2000 (01.09.2000) US

(71) Applicant and
(72) Inventor: **BUCKLEY, Conleth** [CA/CA]; 2651 Colman Street, Ottawa, Ontario K1V 8J7 (CA).

(74) Agent: **SADIK, Achmed, N.**; P.O. Box 908 Station B, Ottawa, Ontario K1P 5P9 (CA).

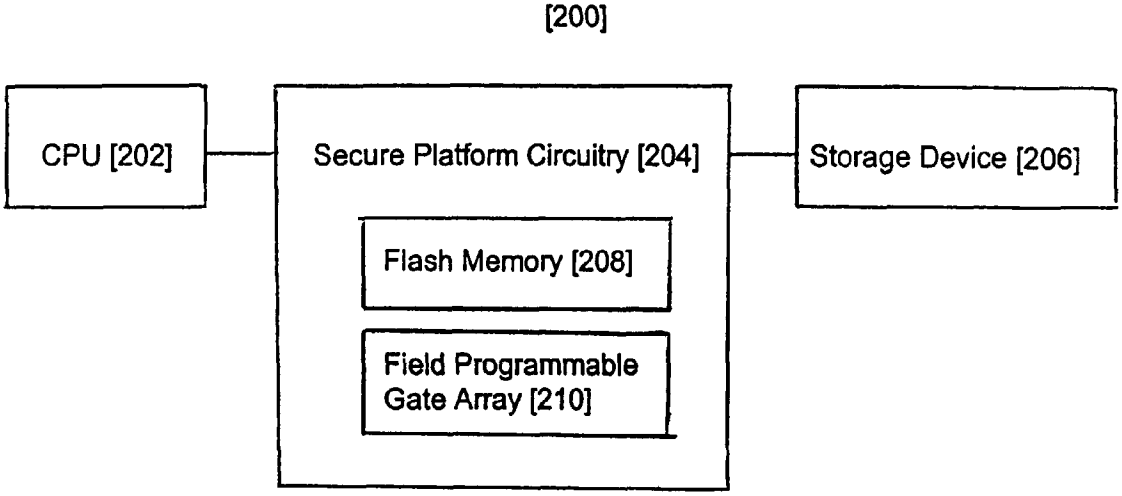
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR PREVENTING UNWANTED ALTERATIONS OF DATA AND PROGRAMS STORED IN A COMPUTER SYSTEM



(57) Abstract: The present invention relates to a method and system for preventing the unwanted alteration of data and programs stored within a computer system. The system employs a field programmable gate array to control access to a storage device. Different profiles can be accessed through the use of passwords. Different profiles provide different control parameters for access to the storage device. The gate array can be reprogrammed from time to time using downloadable electronic files. Security is achieved in the download by using keys and encryption techniques.

WO 02/19064 A2

METHOD AND SYSTEM FOR PREVENTING UNWANTED ALTERATIONS OF DATA AND PROGRAMS STORED IN A COMPUTER SYSTEM

Field of the Invention

The present invention relates generally to computer systems, and more specifically to a method and system for preventing the unwanted alteration of data and programs stored in a computer system.

Background of the Invention

It is well known that computer viruses pose a serious threat to the secure storage of computer data. The term computer virus generally relates to any software code which has been designed to enter a computer and perform an undesired function. Once this code has entered a computer, that computer is said to have been "infected" by the virus.

Computers are most often infected by viruses as a result of introducing software code which has virus code buried within it. This software is typically introduced via an input device such as a disk drive, or via a communication network such as the Internet. Once the software code containing the virus is executed, the virus is activated.

Upon being activated, a virus can perform a wide variety of functions. These functions can consist of relatively harmless functions such as posting an unwanted message on one's monitor or adding additional words to an existing document. These functions, however, can also be very serious and may include occupying all available memory or destroying data and programs stored on the computer or on the hard drive.

Various attempts have been made to try to limit and prevent the damage caused by computer viruses. The most common method of detecting and removing viruses is via anti-

virus software packages. These anti-virus programs, known as virus scanners, detect viruses by searching for binary signatures (patterns of code) of known viruses. Upon detection of a virus the user is notified and the virus is removed.
One limitation of virus scanning software is that the virus protection offered is reactive. That is to say, a virus can only be detected once the binary signature of a particular virus is known and added to the virus-scanning database. Thus, users are not offered any protection against newly created viruses.

A need exists, therefore, for an improved system for preventing the unwanted alteration of computer data and programs.

EXPLANATION OF TERMS:

DHW: Downloadable hardware. The design file that describes the hardware attributes.

DSP: Secure downloadable platform. The hardware where the DHW is loaded.

PKS: A password key system which consists of one or more of the following methods of limiting access to the DSP and/or the storage device protected by the DSP: a series of reads or writes to a series of locations in the storage device; the timing of these writes; challenge/response where the writes depend on the values read. It also includes using any one or more of the following methods: sharing secret knowledge; probabilistic challenges; multi-level passwords; and a one time pad. A feature of this method is that there are many pieces of secret information required to access a storage device.

PASS CODE: Is the string that the users send to the application program, e.g. a password, whereas the PKS is the method that describes the interaction between the application programs and the DSP.

ONE TIME PAD: A sequence of secret numbers which the anti-virus application program uses to identify itself to the DSP.

RANGE: A sequence of blocks for which a particular access applies.

ACCESS: The manner of actions that can occur to the blocks specified in a range; these actions may be one or more of the following: ability to write the block; read the block; translate the read or write from one block to another; to cause an interrupt Control Section: one or more blocks where instructions are passed from the CPU to the DSP.

10

Summary of the Invention

It is therefore an object of the invention to provide a method and system which obviates or mitigates at least one of the disadvantages described above.

One aspect of the invention is described as a method for preventing unwanted alterations of data and programs stored in a computer system comprising the steps of: obtaining a pass code; implementing a profile to prescribe the treatment of at least one command signal in
20 response to the PKS obtained; monitoring data transferred between a CPU and a storage device for a command signal and; responding to at least one command signal based on the implemented profile.

Another aspect of the invention is defined as a system for preventing unwanted alterations of data and programs stored in a computer system comprising: A central processing unit (CPU); a secure platform circuit (DSP); a storage device; the DSP being operable to: obtain a pass code; the DSP being operable to: implement a profile to prescribe the treatment of at least one command signal in response to the pass code obtained; monitor data transferred between the CPU and the storage device; respond to the at least one
30 command signal based on the implemented profile.

Another aspect of the invention is defined as a secure computer platform for down-loadable hardware (DHW) comprising the steps of: monitoring for the reception of a DHW file; determining whether the DHW file is permitted to be installed in response to receiving the DHW file; installing the DHW file in response to determining that the DHW file is permitted to be installed.

Another aspect of the invention is defined as a system for providing a secure computer platform for down-loadable hardware comprising: a central processing unit (CPU); a secure platform circuit (DSP); the CPU being operable to: monitor for the reception of a DHW file; determine whether the DHW file is permitted to be installed in response to receiving the DHW file; the DSP being operable to: install the DHW file in response to determining that the DHW file is permitted to be installed.

The following features, methods and advantages are facilitated by the present invention:

- Prevention of write and/or read to certain blocks of the device or system;
- Control of access to storage devices by means of passwords;
- Detection that the device or system has been "hacked";
- Definition of user profiles to permit different levels of access to device (e.g. disk) or system;
- Use of passwords to: (a) select profiles; (b) permit a series of "writes" to a series of locations in storage device; (c) timing of the "writes"; (d) "writing challenges;
- Providing passwords embedded in random sequences of bytes and check summing the password,
- Password challenges partially driven by the system and partially by the user;
- Providing the concept of, and enabling, downloadable hardware;
- Serialization of product within the downloadable hardware with an encrypted password in order to change configuration;
- Providing a "one time pad" password security;

- Making data invisible by prohibiting block needs to certain areas;
- Making data invisible by returning data from a different block other than the one targeted; and
- Interrupting computer system where access violation is detected.

Brief Description of the Drawings

These and other features of the invention will become more apparent from the following description in which reference is made to the appended drawings in which:

Figure 1 is a block diagram of a computer system as known in the prior art;

Figure 2 is a block diagram of a system for preventing unwanted alterations of data stored in a computer system in an embodiment of the invention;

Figure 3 is a flow chart of a method for preventing unwanted alterations of data stored in a computer system in an embodiment of the invention;

Figure 4 is a flow chart of a method for updating information stored in the protected area of a hard disk in a preferred embodiment of the invention;

Figure 5 is a flow chart of a method for providing a secure computer platform for down-loadable hardware in an embodiment of the invention;

Figure 6 is a flow chart of a method for providing a secure computer platform for downloadable hardware in a preferred embodiment of the invention.

Detailed Description of the Invention

The present invention is directed to a method and system for preventing the unwanted alteration of data and programs stored in a computer system which substantially obviates one or more of the problems due to limitations and disadvantages of the related art.

The reason viruses can perform these unwanted functions is best understood with reference to **Figure 1**. That figure shows a computer system 100 as is well known in the prior art. In particular **Figure 1** shows various input 102, output 104, network communication 106 and storage 108 devices physically linked to a central processing unit (CPU) 110. The CPU 110 performs the various functions of the computer system as specified by various software applications. These functions include directing the operation of the various devices connected to the CPU. A virus, similar to any other software application, is merely a list of instructions which can be carried out by the CPU. Thus, if these instructions include deleting data stored on the computer's hard disk (a storage device), that function will be performed. This is because the instructions provided by a virus are indecipherable to the CPU from those provided by legitimate software applications. Thus, if the CPU is unable to detect the existence of a virus there is nothing to prevent its instruction from being effected.

A block diagram of an embodiment of the present system is shown in **Figure 2**. The system itself 200, includes a central processing unit (CPU) 202, downloadable secure platform (DSP) 204 and a storage device 206. The DSP 204 includes circuitry capable of carrying out the method steps of the present invention. Said circuitry includes flash memory 208 and a field programmable gate array (FPGA) 210. An FPGA is a programmable logic chip which includes numerous arrays of logic block functions and logic gates. The FPGA is programmable to perform a variety of complex functions by modifying the manner in which said gates are interconnected. As will be apparent to one skilled in the art, the flash memory 208 is utilized to store information necessary for

programming the FPGA. Upon the powering of the DPS, the flash memory 208 loads the FPGA 210 with the appropriate circuit design for implementing the methodology of the application and including the invention. In a preferred embodiment of the invention the storage device 206 included in the system is a hard disk drive. The present invention, however, is not limited to employing a hard disk drive and could include any means for storing data including: PDA (personal digital assistant); cell phones; biological storage; floppy disks, CD ROM's or zip disks to name a few.

The present invention, however, is not limited to an FPGA and could also be implemented using an ASIC with some loss of capability.

A methodology for preventing unwanted alterations of data and programs stored in a computer system, in an embodiment of the invention, is shown in **Figure 3**. The methodology initiates at step 300 wherein a pass code is obtained from a variety of possible sources. For example the pass code could be obtained by prompting an end user for pass code via a software system installed on said end user's computer. The user may command an application program to transmit a PSK to the DSP. The DSP only accepts PSK's, not user passcodes. PSK's can be very complex. Similarly, a PSK could be obtained remotely from an end user or an additional computer linked to the CPU via a communication network.

The DSP can be configured to allow access to some parts of the storage device, therefore, no PSK need be received by the DSP unless a change in profile, or other change is desired. A PSK is a sequence of bits which are sent to the DSP at step 302, by transferring a sequence of bytes (groups of binary data) from the CPU to the storage device. As will be explained below, the PSK is used for identifying the various command signals which a CPU may send to the storage device. This list of available command signals which may be sent by the CPU to the storage device is known as a profile. The CPU does not have to be aware that there is a DSP present. The CPU will access the storage device transparently as long as not access is made to a prohibited area.

Upon receiving a pass code the DSP then implements a profile to prescribe the treatment of command signals which can be passed to the storage device based on the PKS forwarded from the CPU 304. This is achieved by enabling the FPGA with a profile implementation associated with the PKS obtained. These various implementations which area associated with the various PKS codes are stored within the flash memory of the DSP or in the storage device.

The methodology continues at step 306 wherein the DSP monitors data being transferred between the CPU 202 and the storage device 206 for a command signal. A command signal can include any write or read signals directed to the storage device. Write and read signals being directions to save to or retrieve data from the storage device respectively.

Upon detecting the transfer of a command signal, the methodology continues at step 306 wherein the DSP responds to the command signal transferred based on the profile implemented within the DSP. In the event that the implemented profile allows for the command to be effected, the DSP responds by allowing the command signal to be passed to the storage device. That is to say, the DSP becomes transparent. If, on the other hand, the implemented profile does not allow for the desired command to be effected, the DSP can respond in a number of ways. First, the DSP can simply prevent the transfer of the command signal to the storage device. Additionally, the DSP can cause an interrupt to be sent back to the CPU to notify the end user, or other computers attached via a communication network, that the desired command is restricted. Alternatively, in the event of a write signal, the DSP could either deny access to the area, or, translate the address ranges to which the write signal is directed to an unprotected area of the storage device. In the event of a read signal, the DSP could translate the address ranges to which the read signal is directed to an unprotected area of the storage device, or could deny it and cause and interrupt.

This methodology can protect against the unwanted alterations of computer data and programs, particularly as the result of computer viruses, in the following ways. First, as

mentioned previously the PKS serves the function of selecting a profile which determines the command signals a particular CPU will be allowed to send to its corresponding storage device. Thus, one can prevent the unwanted alteration of data stored in a computer system by limited the various command signals which are forwarded to a computer's storage device. For example, by merely identifying a range of protected addresses and restricting write signals to these addresses, one can protect against the unwanted alteration of the data stored therein. This is because, as mentioned previously, a virus program operates by initiating a number of unwanted commands to a computer's storage device. If, therefore, a CPU's ability of initiate these commands are restricted, the command signals within a virus will be similarly restricted.

Taking this methodology one step further, it is seen that the PKS concept could also be utilized to minimize any damage caused by a virus where multiple persons or computers share a single storage device. This could be achieved by providing numerous users of a particular computer, or various computers on a network, with distinct pass codes, which implies distinct PKS for the DSP. The different pass codes could then be used to restrict the user's or computer's, access to particular commands and areas with respect to the storage device. The access any particular user would have to the storage device would be stored within the user's or computer's particular profile. Each user may have many profiles and many PKS which will be managed by an application program. For example, the range of addresses to which each end user or computer could write to can be limited.

Additional levels of protection could also be achieved in the system by varying the means by which a pass code is obtained. For example, each attempt to enter a pass code or PKS could be monitored. Therefore, the number of pass code or PKS attempts could be limited to a prescribed value. Thus, snooping viruses, which attempt to bypass security systems by trying all permutations of a particular code, could be guarded against. As an additional level of security the length of the pass code could be increased or the complexity of the PKS could be increased in the event a snooping virus is detected. This would increase the number of permutations and add an additional level of protection against said snooping

viruses. Similarly, snooping viruses could be prevented by requiring the pass code to be entered within a particular time period. That is to say, one could incorporate a timer into the process of obtaining a pass code or PKS. Furthermore, a clock could be incorporated into the process of obtaining a pass code or PKS. Said clock would serve the purpose of limiting the validity of a pass code or PKS to certain time periods. For example said clock could be used to limit the validity of a pass code or PKS to one particular time period (e.g. Jan 1, 2000) or to a recurring time period (e.g. working hours). Additionally, one could incorporate a secondary pass code. This secondary pass code would provide an end user, or computer on a network, with the ability to modify the pass code necessary to access their particular profile. The secondary pass code feature would be beneficial in that a pass code or PKS could be modified in the event of detecting a failed pass code attempt. Multiple passwords could also be used to enable the system. A challenge response system is yet another alternative. A challenge response system operating between the anti-virus application program and the DSP is one by which the program is challenged to return a value when given a number. The challenge response cycle may be repeated a number of times for security. A "one time" pass code could be utilized. That is to say, a different pass code is required each time the circuitry is accessed.

As an additional feature, the circuitry could also be configured to "learn" the locations of programs to protect. This is useful when said system is employed in an operating system which does not know the actual locations of data stored on the computer's storage device. This achieved by writing a start file and an end file marker to the beginning and end of the data which is being protected. Thus the hardware can be made aware of the range of the files to protect.

Referring to **Figure 4** a method is provided for updating the information stored within a protected area of the computer's storage device. The method initiates at step 400 wherein the FPGA is implanted with a profile (P1) which allows one to read part of the contents of the storage device. At step 402 the profile is then changed to one that allows writing to a temporary area within the storage device (P2). A new file (e.g. a new software

application), which is intended to be stored on the storage device, is then written to the temporary area on the storage device 404. A copy of the existing data to be overwritten is then written to the temporary area as well 406. The FPGA is then implemented with the original profile (P1) to ensure that the data copied to the temporary area is correct 408. Once the copied files have been confirmed, the FPGA is implemented with a third profile (P3) which allows one to overwrite files which have been recently copied to the temporary area 410. The new files are then copied to the protected area previously occupied by the files to be overwritten 412. The FPGA is then programmed with the original profile (P1) to ensure the files have been properly updated 414. Finally, the methodology terminates with the FPGA being programmed with P2 such that the old files can be deleted if necessary 416.

Although in the preferred embodiment of the invention these actions take place on a real time basis, it is noted that the invention is not limited in this manner. Rather, the changing of profiles could be delayed, from the action of writing files, such that a snooping virus could not detect that access to the protected area will soon follow. This prevents a snooping virus from writing to the storage device without being detected.

As an additional safeguard, one should note that the process of updating files could be performed by circuitry independent of the CPU. This would, therefore, prevent any snooping program from ever writing to the storage device as said device would only be unprotected when the DSP is performing a copy. This is because the CPU would be prevented from accessing the storage device when the DSP is performing any copies to the storage device. Thus, the unwanted alteration of computer data, particularly those files when are permanent, could not result from a command of the CPU.

Referring now to **Figure 5**, a methodology for providing a secure platform for down-loadable hardware (DHW) is shown in another embodiment of the present invention.

Referring now the **Figure 5**, a methodology for providing a secure platform for down-loadable hardware (DHW) is shown in another embodiment of the present invention.

The methodology presented in **Figure 5** initiates at step 500 wherein the CPU monitors various input or network communication devices for the reception of a DHW file. Upon receiving a DHW file, the CPU then determines whether the DHW file is permitted to be installed in the DSP 502. In a preferred embodiment of the invention this step of determining whether the DHW is permitted to be installed occurs by unitizing a series of Keys and encryptions. An example of a Key and encryption algorithm employed in a preferred embodiment of the invention is described below in further detail with respect to **Figure 6**. The methodology terminates at step 504, wherein the DHW file is installed in the DSP in response to determining that said DHW is permitted to be installed.

The benefit of this methodology is that it ensures that the only DHW files capable of being installed in one's DSP are those which are intended for that particular DSP. In the context of the present invention, i.e. providing virus protection, this methodology is beneficial in that it provides a secure means for updating the implementation stored in the DSP.

The PKS provides a method so that incorrect access to a particular lock within a predetermined number of attempts, or/and time, causes the current PSK to be voided and a new longer identity string and a new longer password assigned. This process is repeated as many times as is desired. This implies that the probability of breaking the lock gets worse with repeated trials and at the same time the probability that the lock could be made un-openable goes down.

From time to time new implementations could be made available to end users. The new implementations could be provided by downloading said implementations via an input or communication network device. This methodology provides a secure platform for installing DHW as it prevents any unwanted DHW from being downloaded directly into the DSP by the CPU.

Each DSP will have its own unique key and serial number.

A flow chart outlining the steps of the key and encryption algorithm utilized in a preferred embodiment of the present invention is shown in **Figure 6**. The methodology initiates at step 600 wherein a completed DHW file is made into an electronic file (F1). A key (K1) is then attached to the electronic file (F1) to create a new electronic file (F2) 602. A key is a secret password which includes a series of characters for restricting an end user's access to an electronic file. The first key (K1) serves the purposes of ensuring each (F2) is unique based on the DSP it is targeted for. This prevents the wrong, or virus contaminated, DHW form reaching the DSP. A second key (K2) is then utilized to encrypt the electronic file (F2) 604. As will be apparent to one skilled in the art of computer data encryption, said file could be encrypted using any number of encryption engines. Each DSP could have a different encryption engine. A third key (K3) is then employed by the DSP to allow the encrypted electronic file F2 to be transferred to a hard-disk protected configuration area (HDCA) via the DSP 606. The HDCA is a temporary storage within the storage device utilized for purpose of storing the DHW file while it is being authorized. This third key (K3) is merely a pass code, utilized by the DSP, allowing the F2 to be written to the HDPA. The encrypted electronic file is then decrypted using the second key (K2) 608. This key would have to be provided to an end user to decrypt the encrypted electronic file (F2). After the file (F2) us decrypted the first key (K1) is extracted from the file leaving only the original electronic file (F1) 610. The first key (K1) is the compared to a key stored with the particular DSP circuitry 612. If the key supposed with the electric file (F1) matches that stored within the DSP circuitry, the file (F1) is installed in the DSP 614. This occurs by the DSP retrieving the decrypted filed (F1) from the HDPA and installing said file (F1) into the flash memory 208. As mentioned previously, the flash memory would then program the FPGA with the new implementations as specified in the electronic file (F1).

One should note that the key referred to in the preferred embodiment is not limited to a single password including a series of characters for enabling a user to access the electronic

file. Said Key could also include a series of transactions wherein various nit (as opposed to bytes) are sent to the DSP form the CPU on a periodic basis.

In the preferred embodiment of the invention the above methodology is used for the purpose of updating DHW for virus protection. One can readily see, however, that said methodology could be easily adapted to provide a secure platform for downloading any DHW.

The preferred embodiment of the hardware of the invention is to build the DSP in the disk storage device circuit board or disk storage assembly.

It will be apparent to those skilled in the art that various modifications and variations can be made in the implementation of the present invention without departing from the spirit and scope of the invention. Thus, it is intended that the present invention cover the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

WHAT IS CLAIMED IS:

1. A method for preventing the unwanted alteration of data and programs stored in a computer system comprising the steps of:

Obtaining a pass code;

Implementing a profile to prescribe the treatment of at least one command singal in response to said code obtained;

Monitoring data transferred between a CPU and a storage device for said at least one command signal;

Responding to said at least one command signal based on said implemented profile.

2. A system for preventing the unwanted alteration of data and programs stored in a computer system comprising:

A central processing unit (CPUT);

A secure platform circuit (DSP);

A storage device;

Said CPU being operable to:

Obtain a pass code

Said DSP being operable to:

Implement a profile to prescribe the treatment of at least one command signal in response to said pass code obtained;

Monitor data transferred between said CPU and said storage device;

Respond to said at least one command signal based on said implemented profile.

3. A method for providing a secure computer platform for down-loadable hardware designs (DHW) comprising the steps of:

Monitoring for the reception of a DHW file;

Determining whether said DHW file is permitted to be installed in response to receiving said DHW file;

Installing said DHW file in response to determining that said DHW file is permitted to be installed.

4. A system for providing a secure computer platform for down-loadable hardward (DHW) comprising:

A central processing unit (CPU);

A secure platform circuit (DSP);

Said CPU being operable to:

Monitor for the reception of a DHW file;

Determine whether said DHW file is permitted to be installed in response to receiving said DHW file;

Said DSP being operable to:

Install said DHW file in response to determining that said DHW file is permitted to be installed.

[100]

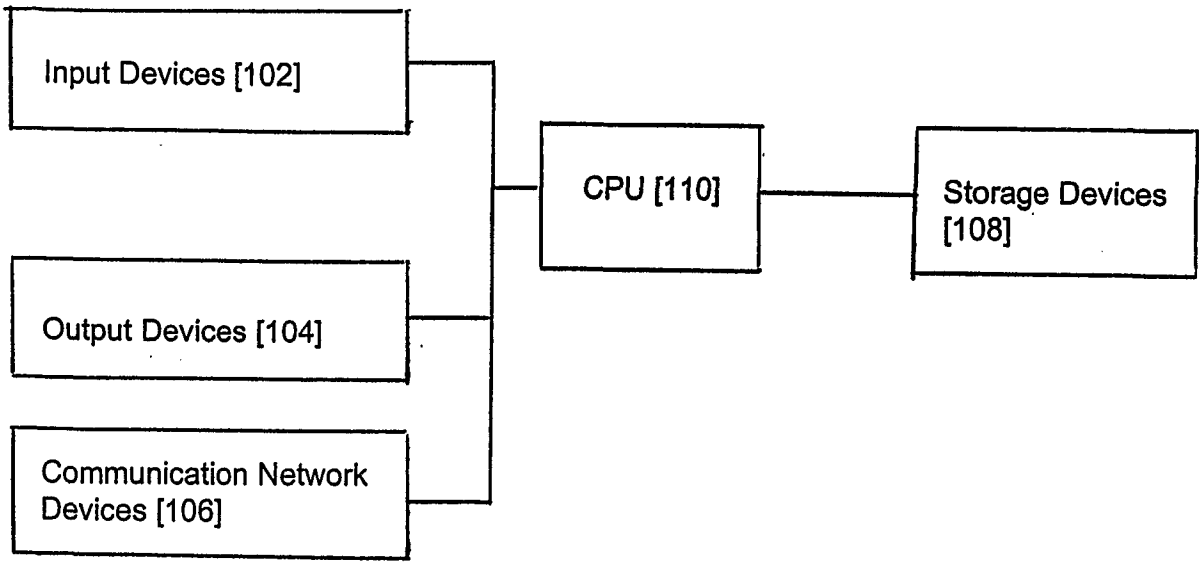

Input Devices [102]
Output Devices [104]
Communication Network Devices [106]
CPU [110]
Storage Devices [108]

FIG. 1

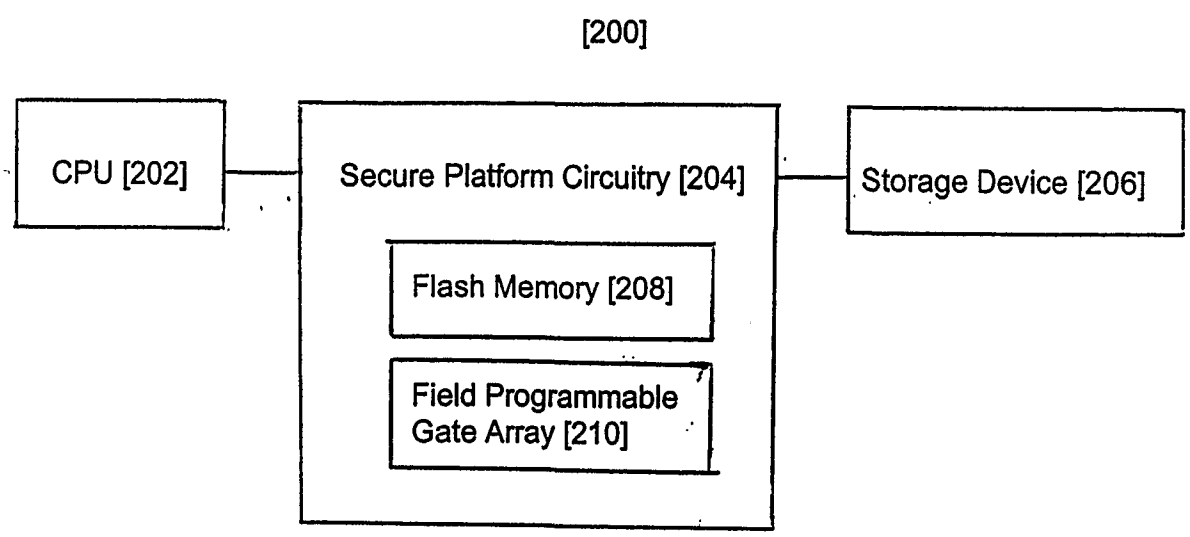

[200]
CPU [202]
Secure Platform Circuitry [204]
Flash Memory [208]
Field Programmable Gate Array [210]
Storage Device [206]

FIG. 2

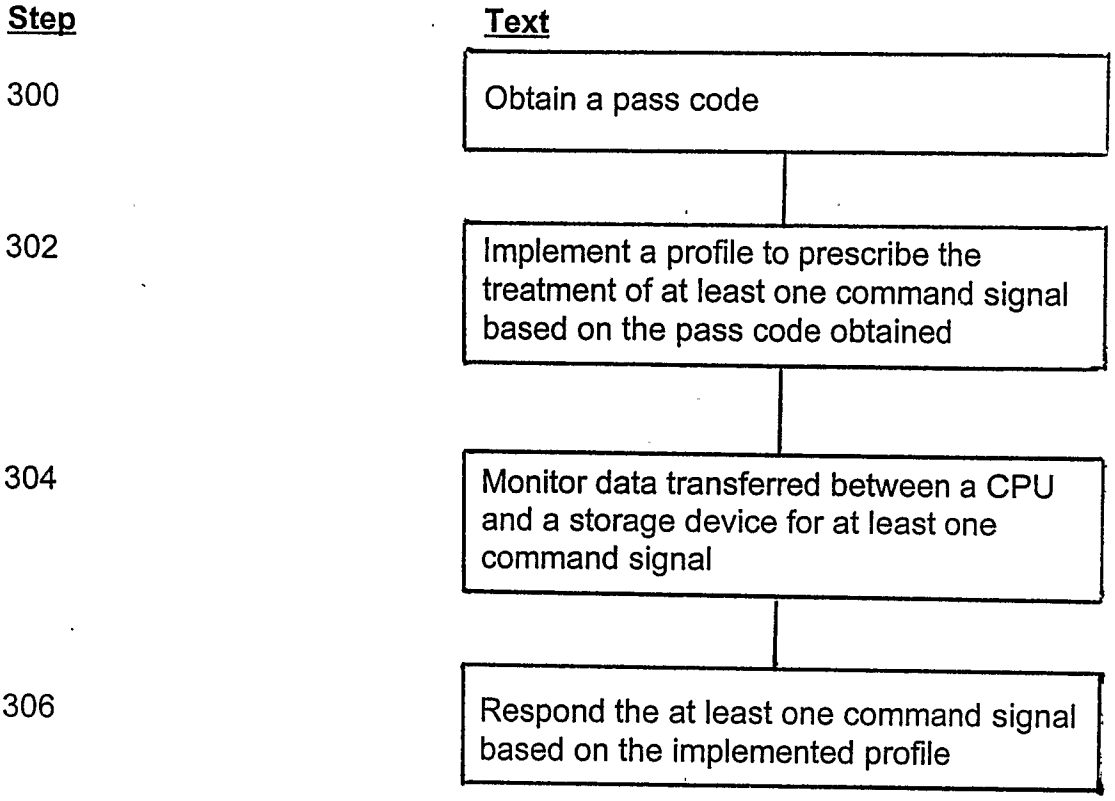

Step
Text
300
Obtain a pass code
302
Implement a profile to prescribe the treatment of at least one command signal based on the pass code obtained
304
Monitor data transferred between a CPU and a storage device for at least one command signal
306
Respond the at least one command signal based on the implemented profile

FIG. 3

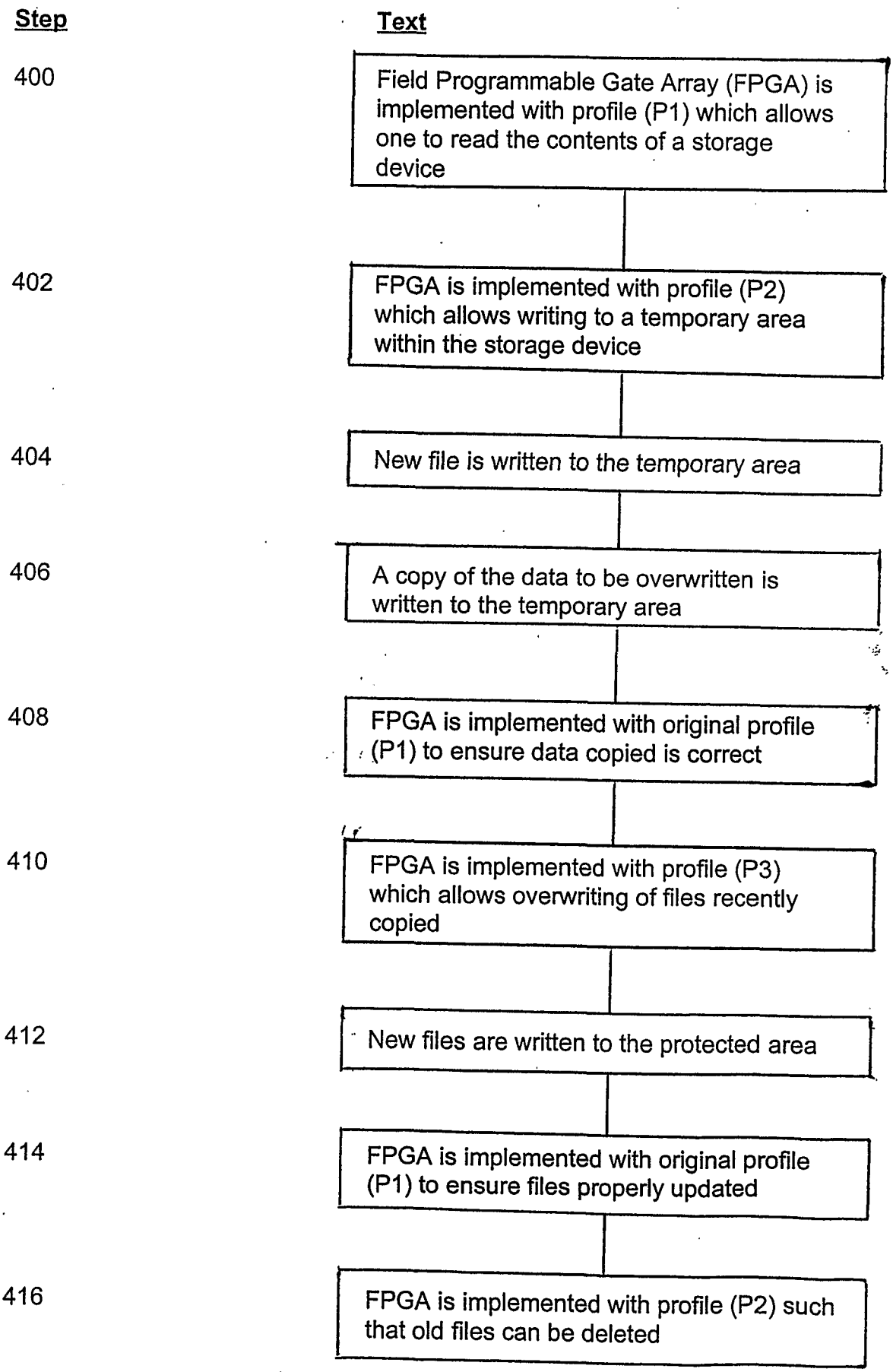

Step
Text
400
Field Programmable Gate Array (FPGA) is implemented with profile (P1) which allows one to read the contents of a storage device
402
FPGA is implemented with profile (P2) which allows writing to a temporary area within the storage device
404
New file is written to the temporary area
406
A copy of the data to be overwritten is written to the temporary area
408
FPGA is implemented with original profile (P1) to ensure data copied is correct
410
FPGA is implemented with profile (P3) which allows overwriting of files recently copied
412
New files are written to the protected area
414
FPGA is implemented with original profile (P1) to ensure files properly updated
416
FPGA is implemented with profile (P2) such that old files can be deleted

FIG. 4

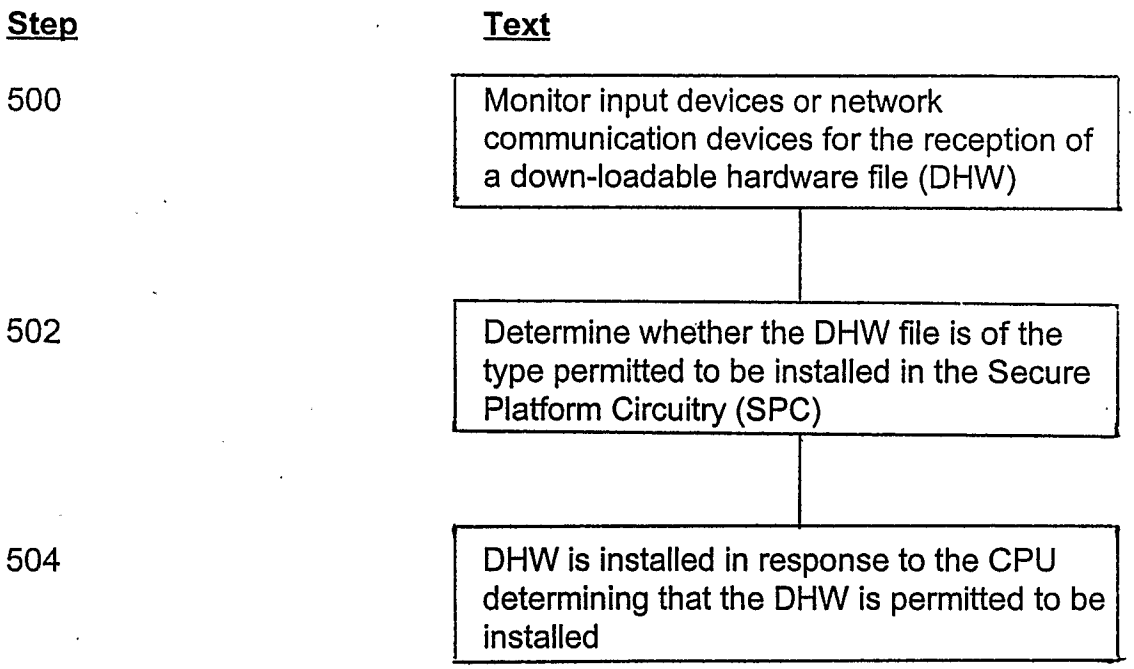

Step
Text
500
Monitor input devices or network communication devices for the reception of a down-loadable hardware file (DHW)
502
Determine whether the DHW file is of the type permitted to be installed in the Secure Platform Circuitry (SPC)
504
DHW is installed in response to the CPU determining that the DHW is permitted to be installed

FIG. 5

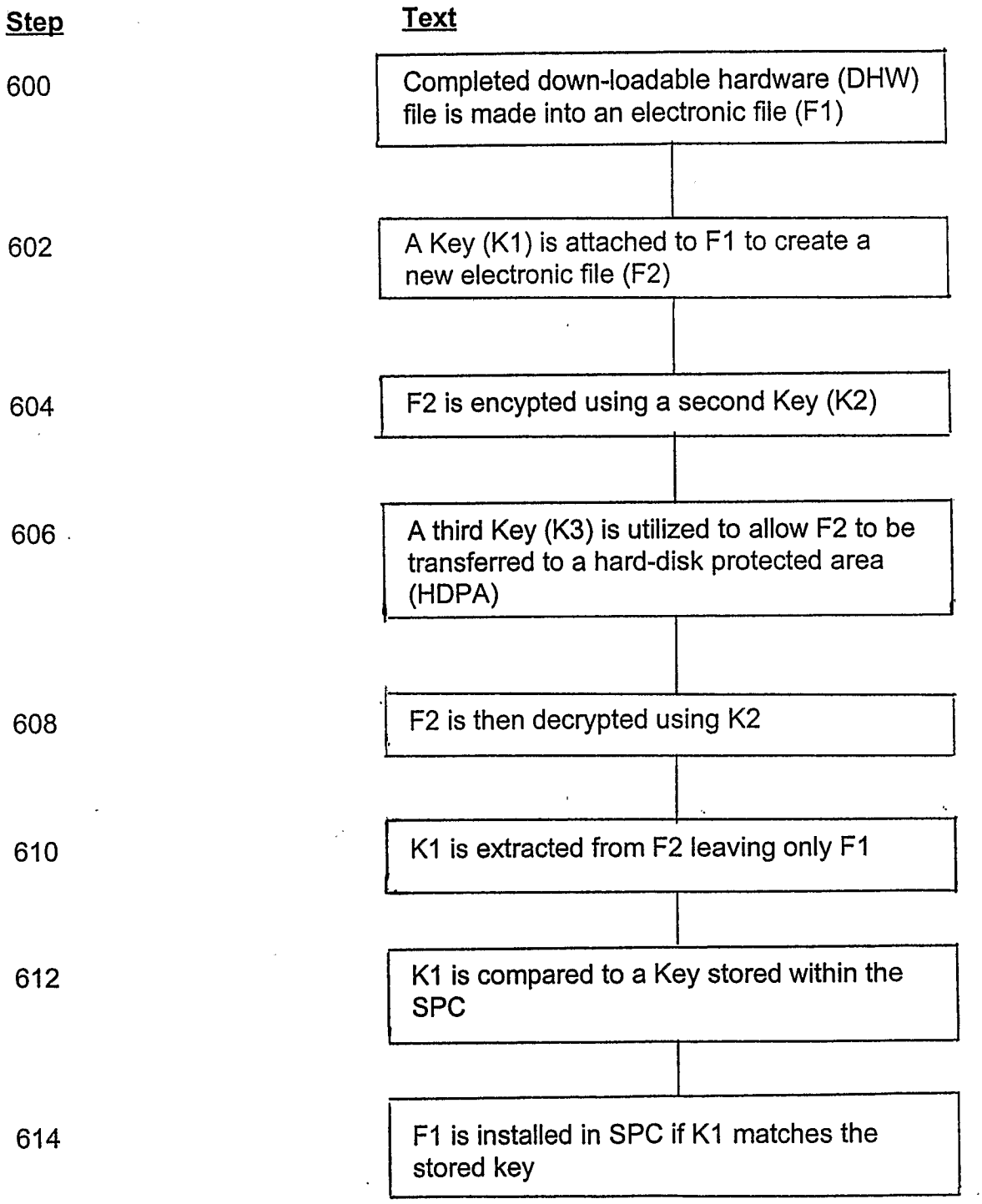

Step
Text
600
Completed down-loadable hardware (DHW) file is made into an electronic file (F1)
602
A Key (K1) is attached to F1 to create a new electronic file (F2)
604
F2 is encypted using a second Key (K2)
606
A third Key (K3) is utilized to allow F2 to be transferred to a hard-disk protected area (HDPA)
608
F2 is then decrypted using K2
610
K1 is extracted from F2 leaving only F1
612
K1 is compared to a Key stored within the SPC
614
F1 is installed in SPC if K1 matches the stored key

FIG. 6